

AMENDMENTS TO THE SPECIFICATION

Please amend the Specification as follows:

On pages 10-11, please amend paragraph [033] as follows:

[033] First, a partitioning method is performed (at 100) by the load utility 20. As noted above, the load utility 20 can open multiple sessions to the database system 10 to perform transactions. Suppose the load utility 20 opens $k \geq 2$ sessions $[[S_i]] \underline{SE}_i$ ($1 \leq i \leq k$) to the database system. If modification operations are randomly distributed among the k sessions, transactions from different sessions can easily deadlock on their X lock requests on the base relations. An X lock is an exclusive lock placed on a table or portion of a table (such as a tuple) when one transaction is updating the table or table portions, to prevent access of the table or table portion by another transaction. The following example invokes a single base relation R and the following four operations (a tuple refers to a row of a table):

O₁: Update tuple t₁ in base relation R.

O₂: Update tuple t₂ in base relation R.

O₃: Update tuple t₂ in base relation R.

O₄: Update tuple t₁ in base relation R.

On page 11, please amend paragraph [035] as follows:

[035] Suppose operations O₁ and O₂ are combined into transaction T₁ that is sent through session $[[S_1]] \underline{SE}_1$. Operations O₃ and O₄ are combined into transaction T₂ that is sent through session $[[S_2]] \underline{SE}_2$. If transactions T₁ and T₂ are executed in the order

T₁ executes O₁,

T₂ executes O₃,

T₁ executes O₂,

T₂ executes O₄,

then a deadlock will occur. This is because both operations O₁ and O₄ require a tuple-level X lock on R for tuple t₁. Also, both operations O₂ and O₃ require a tuple-level X lock on R for tuple t₂.